Bayesian Spherical Wavelet Shrinkage: Applications to Shape Analysis

Xavier Le Faucheur[a], Brani Vidakovic[b] and Allen Tannenbaum[a]

[a] School of Electrical and Computer Engineering,

[b] Department of Biomedical Engineering

Georgia Institute of Technology, 30332, Atlanta, Georgia, USA

ABSTRACT

Multiscale analysis has become a very useful tool in image processing and computer vision. Our work is motivated by the need to efficiently represent 3D shapes that exhibit a spherical topology. This note presents a wavelet based model for shape denoising and data compression. The 3D shape signal is first encoded using biorthogonal spherical wavelet functions defined a 3D triangulated mesh. We propose a Bayesian thresholding model for this type of second generation wavelet in order to eliminate wavelet coefficients that are considered as noise. This way, we are able to to reduce dimension without losing significant information by estimating a *noiseless* version of our shape.

Keywords: Shape analysis, Spherical wavelets, Wavelet shrinkage, Bayesian model

1. INTRODUCTION

Shape analysis has become a very important part of image processing and computer vision. To represent shapes in a parametric form, different methods have been developed so far. Recently, multiscale analysis has brought some improvement in terms of spatial decomposition and the use of wavelets has been proposed for shape representation.

In this paper, we use the work done by Nain et al.[1] to represent our shapes. This consists in encoding our 3D signal with the spherical wavelets proposed by Schroeder and Sweldens[2] . This type of second generation wavelet allows us to represent shapes within a few scales of analysis. In[1] , Nain et al. develop their analysis over a given dataset of shapes by encoding their variation from the mean shape using spherical wavelet. Then, they retain the main variations by truncating some of the wavelet basis functions via a power analysis. They actually remove the lowest basis functions whose cumulative contribution is lower than a chosen percentile of the total power. This allows to reduce dimensionality in the wavelet space and to model the given shape with a smaller number of modes of variation and with no significant loss of information.

Here, for any shape with a spherical topology and given its spherical wavelet decomposition, we want to remove those of its wavelet coefficients that are considered as *noise* without losing significant information on the shape. Wavelet thresholding and wavelet shrinkage are well-understood and have been widely researched for traditional types of wavelet decompositions but not much explored for the second generation wavelets. Donoho et al.[3] proposed a selective wavelet reconstruction model for signal extraction from noisy data by defining a threshold (universal, SURE) and apply either a soft or a hard shrinkage to their wavelet coefficients. Bayesian wavelet shrinkage has been first proposed for classical wavelets by Vidakovic[4] . By applying a Bayesian thresholding to our wavelet coefficients, we are able to efficiently denoise our shape signals and drastically reduce dimension. Also, we introduce some neighboring information into our model to make it more consistent and our resulting shapes more regular.

Further author information: (Send correspondence to Xavier Le Faucheur)
Xavier Le Faucheur: E-mail: xavier@gatech.edu, Telephone: 404-385-5062
Brani Vidakovic: E-mail: brani@bme.gatech.edu, Telephone: 404-385-7246
Allen Tannenbaum: E-mail: tannenba@bme.gatech.edu, Telephone: 404-894-0290

In this note, we will first present our shape model, making use of the work of Nain. Next, we will first recall the spherical wavelet definition and its implementation scheme. Then we describe the wavelet shrinkage model we use for denoising. Finally, we show some results on a dataset of left hippocampus shapes and compare our result to more basic methods like universal thresholding.

2. THE SHAPE MODEL

A shape will consist of a $3D$ mesh with N vertices. The shape signal is represented by a $3N$- vector that we apply our wavelet transform to. The goal of our method is to remove wavelet coefficients that we actually consider as noise. Let us then propose this model of shape representation:

$$\vec{y} = \vec{f} + \vec{\varepsilon}, \tag{1}$$

where $\vec{y}$ is the observed shape signal, $\vec{f}$ a $3N$ vector containing the shape signal (x, y, z) we want to estimate at the N vertices, and $\vec{\varepsilon}$ has $3N$ normal i.i.d entries.

From the linearity of the wavelet tranform, we obtain a similar equation for the shape representation in the wavelet domain:

$$\vec{d} = \vec{\theta} + \vec{\epsilon}. \tag{2}$$

where $\vec{d}$, $\vec{\theta}$ and $\vec{\epsilon}$ are the wavelet coefficient vectors obtained after applying the wavelet transform W to $\vec{y}$, $\vec{f}$ and $\vec{\varepsilon}$ respectively. In the vectors of coefficients the first N entries correspond to the x-coordinates of the shape, ranked from coarse resolution to high resolution, the next N entries to the y-coordinates and the last N ones to the z coordinates. Given a set of coefficients $\vec{d}$ that we define as noisy, we want to estimate the wavelet coefficients belonging to the noiseless signal $\vec{\theta}$. Even if our wavelet transform is not completely orthogonal, but biorthogonal, we can show that it is reasonable to consider the components of $\vec{\eta}$ as i.i.d normal random variables.

3. SPHERICAL WAVELETS

The shapes we are using are topologically spherical. Thanks to a one-to-one mapping from our surfaces to the sphere, we equip them with a multiresolution mesh created by recursively subdividing an initial polyhedral[1] . This is done by adding a new midpoint at each edge and splitting each triangle into 4 new triangles. There, we build our wavelet system on the equivalent sphere after Sweldens and Schroeder's work. A spherical wavelet basis is composed of functions defined on the sphere that are localized in space and characteristic scales. We use this type of wavelet to encode the (x, y, z) coordinates at each vertex, using this subdivision scheme.

These spherical wavelets are second generation wavelets and are based on a *lifting scheme* which allows one to custom design the wavelet filters and to have a much better representation of the signal.[2] The basis functions have a lot of desirable properties and are adapted to very general domains of definition. They are not simple dilations and translation of a mother basis.

The computation of the wavelet coefficients is done for each level of decomposition, starting at the finest one. The scheme that is used to compute these coefficients is based on the regular multiresolution mesh definition. A wavelet basis function at a given vertex uses neighboring points according to the *butterfly scheme* (Figure 1).

Now given the number of coefficient used to represent our shape in the wavelet domain, we can assume that a good part of it can be ignored and considered as noise signal. This motivates our wavelet shrinkage.

4. HYPOTHESIS TESTING AND COEFFICIENT DISTRIBUTIONS

Our parsimonious coefficient selection is based on a Bayesian hypotheses testing and we want to have an optimal estimation for θ. We set to zero the coefficients that are just noise and keep the meaningful ones. Let us set the null hypothesis as follows: $H_0 : \theta = 0$. If this hypothesis is rejected, we will keep $\theta = d$ for the considered coefficient. The coefficients are clustered by level of decomposition and we run our analysis level wise. Let us denote by j the level of decomposition, by k the index of the coefficient in this level and by c the coordinate (x, y or z) of the signal we are considering. We will consider the 3 dimensions of our shape signal as independent from each other and will deal with each coordinate separately. For more convenience indices are dropped.

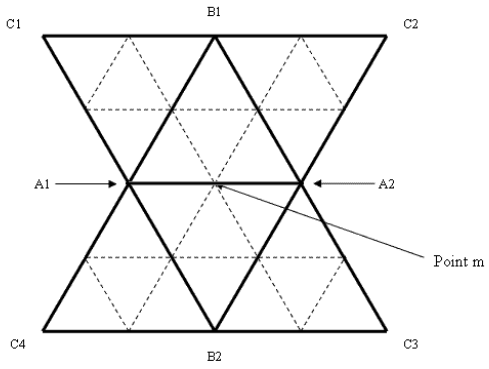


Figure 1. "Butterfly" neighboring system for a point m at a level j: its neighbors A1, A2, B1, B2, C1, C2, C3 and C4 are vertices from coarser levels and are used to compute its wavelet coefficient value and basis function. Dashed lines form the new triangles of level j.

4.1 Marginal Likelihood

Since we assume the noise to be Gaussian, we can write the marginal likelihood as follows:

$$d|\theta \sim N(\theta, \sigma^2) \tag{3}$$

where the noise variance will be estimated for each level of decomposition. This estimation can be done using a power spectrum analysis in the wavelet domain. For each level and for each coordinate of the shape signal, we compute the following:

$$P_d(j) = \log(\frac{1}{N_j} \sum_{k \in J} d_{jk}^2), \tag{4}$$

where j indicates the level and N_j the number of coefficients at this level. This power function empirically evaluates the variance of the observed coefficients at each level by approximating $\log(E[d_j^2])$. Also, we observe empirically that we can reasonably approximate the wavelet coefficient log-periodogram P_θ of a *noiseless* shape signal by an affine function. This assumption is commonly used for classic wavelets. Here, we assume that coarser levels are not significantly affected by noise since they represent low frequency variations. We then use P_d values to evaluate P_θ at coarser levels. Using a linear regression to estimate the slope of P_θ based on coarser levels, we are able to estimate P_θ at finer levels (Figure 2). From there, we can compute the estimated noise variance for each one of the finer levels:

$$\sigma_j^2 = \exp(P_d(j)) - \exp(P_\theta(j)) \tag{5}$$

4.2 Prior distribution

We have to pick a prior distribution for the wavelet coefficients that would be consistent and adapted to hypothesis testing. For classical wavelets, Vidakovic proposed to set up the prior distribution of coefficients as the weighted mixture of a point mass at zero with an other distribution ξ:

$$\theta \sim \pi_0 \delta_0 + \pi_1 \xi(\theta) \tag{6}$$

We will choose the latter from the double exponential family:

$$\xi(\theta) = \frac{1}{2\tau} e^{-|\theta|/\tau} \tag{7}$$

The main point here is about estimating the parameter τ. From (3) and because of the independence between the noise and the signal we want to estimate, we have:

$$\sigma_d^2 = 2(1 - \pi_0)^2 \tau^2 + \sigma^2, \tag{8}$$

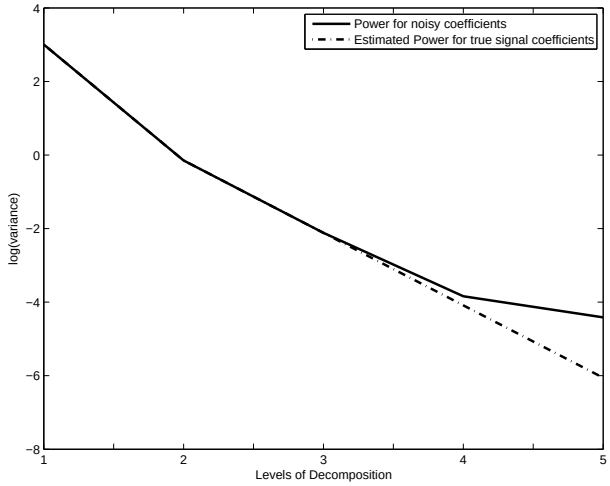


Figure 2. Power spectrum for x-coordinate coefficients with respect to the level of decomposition. Here we assume that the three first levels are not affected by noise. P_θ values (the dashed curve) and P_d values (solid) are then assumed to be equal at those levels

where σ^2 is the estimated noise variance for a given level of decomposition and σ_d^2 is the variance of the *observed* coefficients from this level.

By using (5) to estimate the noise variance and plugging this value in (8), we are able to find a good estimation for the parameter τ at each of the finest level of decomposition.

4.3 Hypothesis testing

Now, at each of the finest levels and for each coordinate separately, we test our hypothesis H_0 versus $H_1 : \theta \neq 0$. We will considered as *to be shrunk to 0* the coefficients that satisfy (Bayes rule):

$$\frac{P(H_0/d)}{P(H_1/d)} = \frac{\pi_0 e^{-(\|d\|^2/\sigma^2)}}{\pi_1 \int e^{-(\|d-\theta\|^2/\sigma^2)}\xi(\theta)d\theta} > 1 \tag{9}$$

The 3 dimensions are processed in a complete independent way.

5. HYPOTHESIS TESTING WITH NEIGHBOR INFORMATION

5.1 Inter-level dependence

Our objective here is to obtain a shape representation that would combine smoothness and precision. We want to keep as much information as possible while removing irregularities that correspond to high frequency regions. With the method presented above we would basically just keep high valued coefficients and shrink the small ones. Here, by taking into consideration neighboring points from coarser levels, we can determine if it makes sense to really keep a coefficient or shrink it. Thus, if a high-valued coefficient from finer levels has low-valued neighbors, then this coefficient is more likely to be considered as noise and then shrunk. This is why we introduce some neighboring information into our thresholding model.

This neighboring information will be contained in our prior distribution by making the weights π_0 and π_1 neighbor dependent. The main point here is about determining the type of neighboring system we should use. The scheme used to build our discrete biorthogonal wavelets is based on our multiresolution mesh and the computation of a wavelet coefficient at an arbitrary vertex uses information from coarser level neighbor points. On Figure 1, we see the neighbor points that are used in our bases. Considering this scheme and a "new point" k at level j, the most *influencial* neighbors will be the points $A1, A2, B1$ and $B2$.

The weight π_0 will be a function of the wavelet coefficients at the neighboring points. In practice, for a wavelet coefficient d, we will pick its neighboring points, normalized their wavelet coefficient values since they can be from different coarser levels, and average these values to end up with a value c, which represents the "strength" of the neighborhood. We propose to model it as follows:

$$\begin{cases} \pi_0(neighbors) = K & if \quad c \leq (sign(d)) \cdot T \\ \pi_0(neighbors) = \alpha \cdot e^{-\beta \cdot (sign(d)) \cdot c} & if \quad c \geq (sign(d)) \cdot T \end{cases} \tag{10}$$

where T,α, and β are free parameters to set up and on those will depend the number of shrunk coefficients. K will basically be close to 1. We see here that when a coefficient and its neighbors have opposite signs, the coefficient is very likely to be shrunk ($\pi_0 = K$). We could also consider that a high-valued coefficient with a high-valued but opposite signed neighborhood should be kept. We then consider as "to be shrunk to 0" a coefficient that satisfies:

$$\frac{\pi_0(neighbors) e^{-(\|d\|^2/\sigma^2)}}{\pi_1(neighbors) \int e^{-(\|d-\theta\|^2/\sigma^2)} \xi(\theta) d\theta} > 1 \tag{11}$$

As mentioned earlier, we assume that noise only affects the finest levels of decomposition. We will then have a recursive shrinkage method, starting at the lowest level that is affected by noise, shrink coefficients from this level, use those new coefficient values for the next step, and keep going up to the finest level.

5.2 Double shrinkage

Because of the configuration of certain shapes and the triangulation-dependent localization of certain vertices on the shape, it may happen that neighbors from coarser levels give meaningless information about finer level points. That is why we propose here to apply a *second shrinkage phase* to our model, which will be based on neighbors from the same level (*intra-level dependence*). We show how this method improves our thresholding, especially for noisy shape signals.

As mentioned in the previous section, we can look at the 3 coordinates together and shrink the 3 coefficients $d_{jkx}, d_{jky}, d_{jkz}$ if and only if the three of those are considered as "to be shrunk".

6. RESULTS AND DISCUSSION

6.1 Experimental Protocol

We use a dataset of 25 left hippocampus shapes. The multiresolution mesh is created by recursively subdividing an initial tetrahedron. There are 5 levels of subdivision and we get 2562 vertices to represent our shapes. We use the work of Nain for shape registration and remeshing with triangle area preservation.

At each vertex, the shape signal (x, y, z) is encoded using the spherical wavelet transform. The wavelet basis functions are built on the sphere itself and the coefficients are clustered by level of decomposition.

We will consider as *noisy* the last two levels of decomposition and will apply our thresholding to these levels only. For the parameters in (10), we choose $K = .95$, $\alpha = 2$ and $\beta = 6$.

After shrinking the wavelet coefficients, we apply the reverse transform to recover our new shape signal $(x_{new}, y_{new}, z_{new})$ and compute the error at each point by comparing our result to our original shapes.

We compare our two shrinkage models to the *universal* hard thresholding as developed by Donoho[3] for classical wavelets using a threshold λ defined levelwise as $\lambda = \sqrt{2 \log N} \cdot \sigma$ where N is the number of coefficients for the given level and σ the noise standard deviation.

6.2 Data Compression

First, we apply our shrinkage method to our original shapes and see if we can get rid of lots of coefficients without losing any significant information. Table 1 presents the average number of coefficients that have been removed for each dimension and the reconstruction error. As shown in that table, we were able to remove a lot of coefficients (around 2000) and still represent shapes with accuracy. Since our shapes here are not affected by noise, our Bayesian method -with or without neighbor information- does not outperform universal thresholding, but can at least compete with it.

6.3 Denoising and Data Compression

Here we test our method on noisy shape signals. We add Gaussian noise to our original shapes and try to reconstruct the original shape with accuracy. The results are presented below. Figure 4 shows the resulting meshes after applying the three different techniques (4-c, 4-d, 4-e). We can observe that our Bayesian model allows us to recover the original shape with precision. The smoothness of the shape in 4-e emphasizes the improvement brought by our *inter- and intra-level dependent* shrinkage. Table 2 presents the average number of coefficients that have been removed and the average reconstruction error. On Figures 5 and 6, we observe the distribution of the reconstruction error for the three techniques. All these results show how the neighbor-based Bayesian thresholding improves our shape representation in terms of compression, precision and smoothness.

6.4 Discussion

Even though our method tends to outperform classic thresholding for noisy shape signals, one main concern remains the noise variance estimation. On this estimation depends the estimation of parameter τ (8). It can actually occur that the graphs in (Figure 2) exhibit some irregularities which would lead to pretty bad inaccuracies for noise variance estimation. Also, π_0 has a very big influence on the strength of the shrinkage. This will actually have two opposite effects onto our shrinkage. High values for π_0 will tend to strengthen the shrinkage on one hand and to soften it on the other one since it intervenes in the definition of τ (equation 8). However the robustness of our model makes those parameter estimations much less important than in the classic universal thresholding, where the choice of the threshold is very crucial and the results more arbitrary.

7. CONCLUSION AND FUTURE WORK

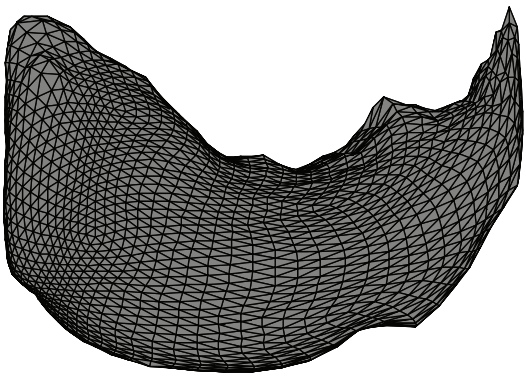
Using this spherical wavelet tool, we are able to represent shapes with precision within a few scales of analysis. Here we have shown that a well-adapted statistical thresholding method can help us remove a lot of spherical wavelet coefficients that are not needed in our shape representation. This is more efficient than the classic universal thresholding method that is very sensitive to the choice of the threshold, and then to the noise variance estimation.

The method we developed is pretty robust to noise and information from neighbors turns out to be a very useful tool in our shrinkage model. However this method remains very dependent on some hyper-parameter estimation.

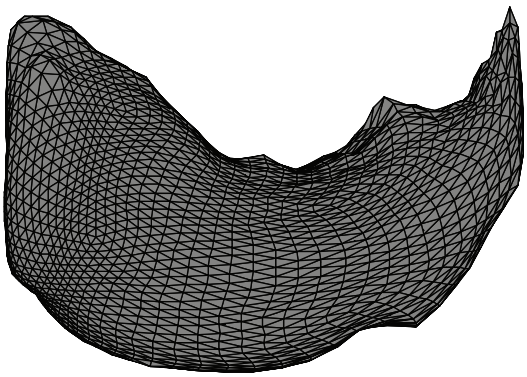
In a future work, we will apply this to shapes corrupted by non-Gaussian noise. Noise variance estimation also remains a challenging task.

REFERENCES

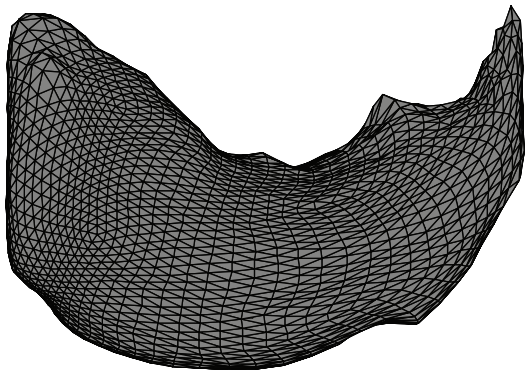
1. D. Nain, S. Haker, A. Bobick, and A. Tannenbaum, "Multiscale 3d shape representation and segmentation using spherical wavelets," *Trans. on Medical Imaging* **26**, pp. 598–618, April 2007.
2. P. Schroeder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," *Computer Graphics* **29**, pp. 161–172, 1995.
3. D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Tech. Report, Statistics, Stanford* , 1992.
4. B.Vidakovic, "Nonlinear wavelet shrinkage with bayes rules and bayes factors," *Journal of the American Statistical Association* **93**(441), pp. 173–179, 1998.



(a) original shape

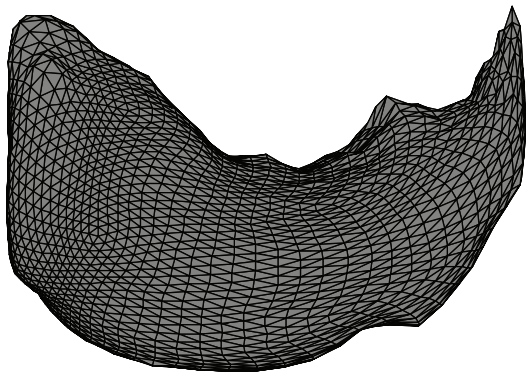


(b) shape after universal thresholding

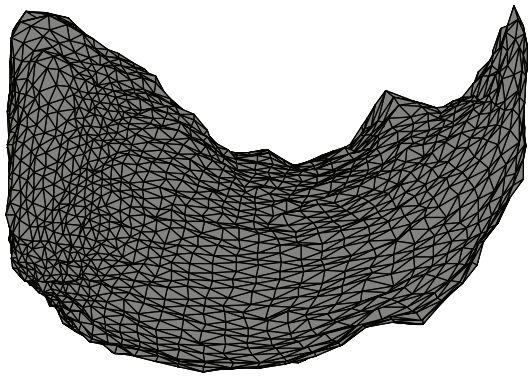


(c) shape after Bayesian thresholding with neighbor information

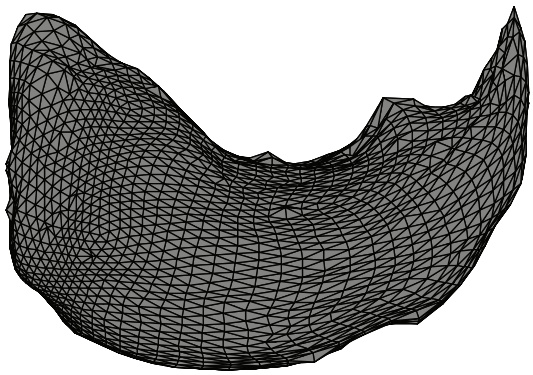
Figure 3.



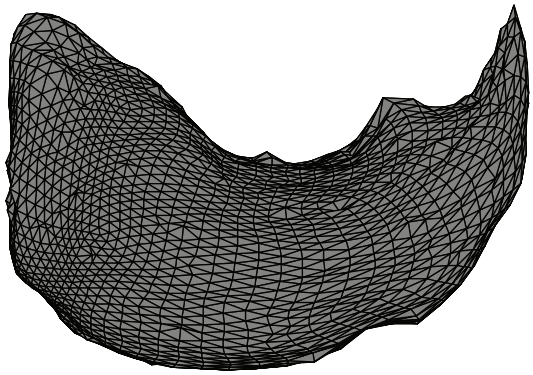
(a) original shape



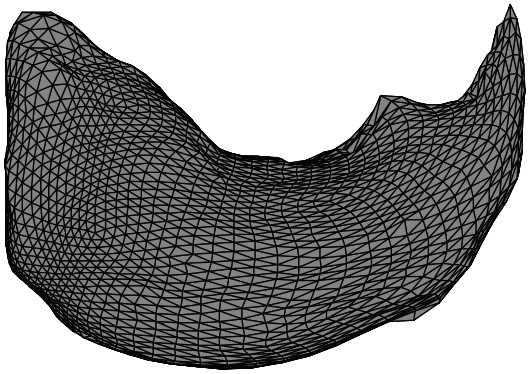
(b) noisy shape with $\sigma = 0.1$



(c) shape after universal thresholding

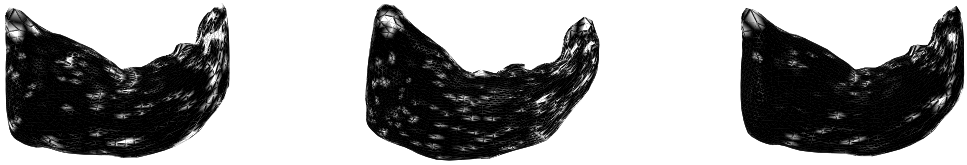


(d) shape after Bayesian thresholding without neighbor information



(e) shape after Bayesian thresholding with neighbor information

Figure 4.

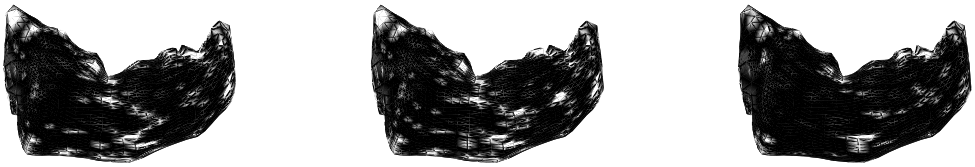


(a) universal thresholding

(b) Bayesian thresholding without neighbor information

(c) Bayesian thresholding without neighbor information

Figure 5. Error reconstruction for shape 7. White is error greater than 1% and black is error less than 1%



(a) universal thresholding

(b) Bayesian thresholding without neighbor information

(c) Bayesian thresholding without neighbor information

Figure 6. Error reconstruction for shape 8. White is error greater than 1% and black is error less than 1%

	universal	Bayesian without neighbors	Bayesian with neighbors
average number of shrunk coeffs over the 3 dimensions	1970	2160	1938
average mean error (%)	0.54	0.48	0.38
average max error (%)	1.83	1.64	1.44

Table 1. Result comparison for data compression on original shapes. We compare the results for the basic thresholding method (column 1), the Bayesian thresholding without (column 2) and with (columns 3) any neighbor information. For each of those methods we compute the average number of coefficients that are thresholded for each dimension (first 3 lines) and the average of the mean and the maximum error (in percentage of the shape bounding box)

	universal	Bayesian without neighbors	Bayesian with neighbors
average number of shrunk coeffs over the 3 dimensions	2278	2238	2170
average mean error (%)	0.77	0.77	0.71
average max error (%)	3.22	2.94	2.7

Table 2. Result comparison for data denoising and compression for noisy shapes. We compare the results for the basic thresholding method (column 1), the Bayesian thresholding without (column 2) and with (columns 3) any neighbor information. For each of those methods we compute the average number of coefficients that are thresholded for each dimension (first 3 lines) and the average of the mean and the maximum error (in percentage of the shape bounding box)